intel

APPLICATION NOTE

AP-318

June 1988

Intel's 87C75PF Port Expander Reduces System Size and Design Time

TERRY KENDALL
MICROCONTROLLER PERIPHERALS
INTEL CORPORATION

Order Number: 292048 001

INTEL'S 87C75PF PORT EXPANDER REDUCES SYSTEM SIZE AND DESIGN TIME

CONTENTS PAGE

INTRODUCTION

What's tho driving factor in your embedded control application? Board space? Reliability? Power? Design time? Manufacturing simplicity? Cost?

What if a single component helped you achieve smaller board size, higher reliability, lower power, faster design time, simplified manufacturing, and lower cost? Intel's 87C75PF is the first in a family of microcontroller peripheral port expander products. This application note will show how the 87C75PF significantly reduces chip count and greatly simplifies system design. The 87C75PF data sheet has detailed device information.

Intel's early microcontrollers had obvious benefits over previous alternatives — a high degree of system integration. The most common microcomputer functions — CPU, ROM, RAM, I/O ports, timers/counters, address decoding, etc. — were combined onto a single chip. Upgrades and proliferations have grown significantly since those early days. Four-bit and 8-bit controllers are the most widely used, with 16-bit versions, spearheaded by Intel's 8096 family, beginning their exponential growth.

The most sought after microcontroller improvement is additional program memory. 8- and 16-bit controllers are optionally equipped with 4K or 8K bytes of ROM or EPROM. This is sufficient memory for about half of embedded applications.

The remaining applications use off-chip EPROM. One reason, of course, is to increase system memory; typically to 16K- or 32K-bytes. Another is to provide flexibility for code that changes frequently. In other applications, generic boards or multi-use modules can be manufactured and custom-programmed for special configurations. For example, a single robot control module can be manufactured. Identical robots can be configured to perform various factory tasks.

8- and 16-bit microcontrollers accommodate external-memory expansion. Controllers sacrifice two 8-bit I/O ports to supply address and data lines to peripheral components. Unfortunately, expanded-memory modes violate two embedded-control objectives, maximizing I/O capability and reducing chip count (or board size). Usually, systems that need more memory are also I/O intensive. Traditional memory-expansion/port-recovery schemes use multiple chips. Memory, address latches, port latches, transceivers, address decoders, and glue chips turn a single-chip uC system into a multiple-chip conglomeration.

THE MULTIPLEXED BUS

To achieve small board size, embedded control systems require minimum chip count and chips that occupy small footprints. Embedded controllers use multiplexed address/data buses to achieve both. An 8051 controller, for example, shares its lower eight address pins with its 8-bit data.

Every memory access requires two cycles — one for address, one for data (see Figure 1). The controller's first cycle places a 16-bit address on the bus. It holds the upper eight bits constant throughout the access. It presents the low-address byte just long enough for an external latch to capture it. The latch and controller's upper bus then supply the 16-bit address to external devices for the remainder of the memory access. The controller's data cycle transmits or receives data on its multiplexed lower address/data pins. The multiplexed bus minimizes the controller's pin count and the system's board traces.

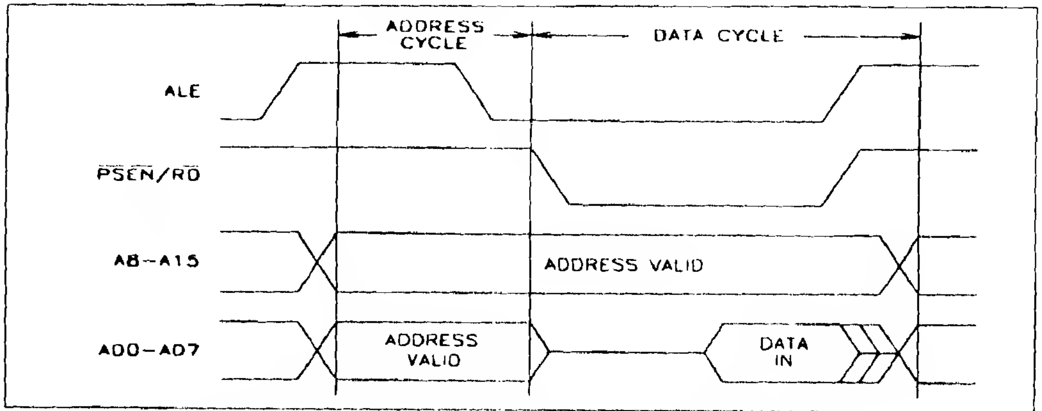

Figure 1. Every microcontroller memory access requires two cycles.

WHY A PORT EXPANDER?

Single-chip microcontroller solutions are quickly giving way to multiple-chip, high-end solutions. Embedded control applications often require more program memory than the microcontroller's on chip memory. Sometimes, code flexibility is needed. The 87C75PF's 32K byte EPROM dwarfs any microcontroller's on-chip memory.

In the near future, microcontroller chip-sets — controller and peripheral — will make up most embedded control applications. The controller will contain features that must be coupled closely to its CPU. The peripheral chip will provide memory and I/O functions.

Controller and peripheral-chip costs will be more balanced. The chips will share complexity, which equates to cost. Two smaller, less complex chips will cost less than one huge controller chip, resulting in lower total system cost.

Typically, adding external functions to microcontrollers requires many chips and substantial board space. Address latches, memory, port recovery, and glue chips require far more space than a single-chip microcontroller. System reliability and performance are degraded. Design and manufacturing are more complicated.

Intel's high-performance 87C75PF Port Expandor doesn't compromise designers' goals to create reliable, minimum chip systems. Its single chip, no glue interface simplifies design and manufacturing while increasing performance and reliability — in the smallest possible board space.

A TYPICAL SYSTEM

Intel's 8051 microcontroller architecture is the most widely used. Many variations are available with enhanced I/O features and various amounts of memory. Intel's 80C31 is a non-ROM, CHMOS version of the 8051. It will help illustrate the 87C75PF's benefits over typical multiple-chip uC solutions.

Figure 2 shows a typical expanded microcontroller system. Whenever memory-mapped devices are connected to a microcontroller, two 8-bit ports lose their I/O functions to become address and data pins. Figure 2 shows port-reconstruction devices, a 256K-bit EPROM, and glue chips that make up an embedded control system. Nineteen chips are required!

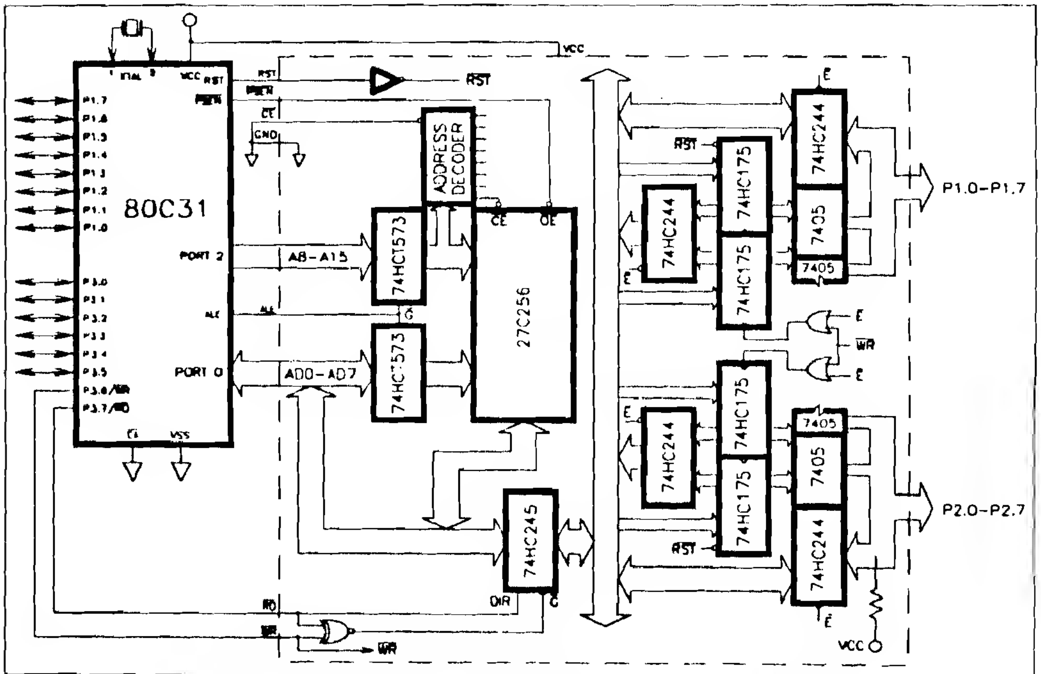

Figure 2. Many descrete chips provide EPROM and port expansion.

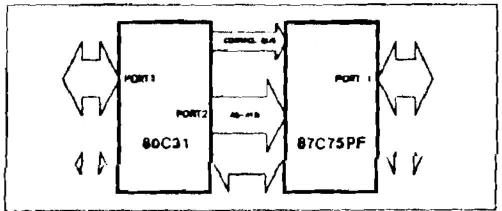

Figure 4. A "no-glue", two chip 87C75PF system.

inventory, incoming inspection, testing, manufacturing, board size, and rework costs.

Reliability also has significant value — to you and your customers. Customers demand products that work properly — forever. Reworked products waste time and money, increase the cost of every unit you ship, and ruin your company's reputation. The best way to increase reliability is to eliminate system components.

Simplified manufacturing saves time and money while increasing reliability. One factory-tested, integrated-function chip is much easier to place on a circuit board and is far more reliable than myriad discrete chips. Every solder joint is a possible failure point. A single chip reduces potential failure points from hundreds to a few.

87C75PF ARCHITECTURE

The 87C75PF Port Expander's features include:

- Two 8 bit I/O ports
- 32K x 8 EPROM
- Two 64K-byte memory planes
- Special Function Registers
- Device-configuration registers
- "No-glue" controller interface
- Low-power, Low-noise CHMOSII-E
- Quick-Pulse Programming™ Algorithm
- In system programmability
- 40-Pin CERDIP, 44-Lead PLCC packages

Two Ports

The 87C75PF has two 8-bit bi-directional I/O ports. Port 1 has open-drain outputs and port 2 has quasi-bi-directional (resistor pull-up) outputs. Each port is individually addressable with separate port-latch and port-pin addresses. Typical of quasi-bi-directional ports, they are always in output mode but can be used as inputs by simply writing logic "1s" to their latches.

Relocatable EPROM

The EPROM has 262,144 bits organized as 32K 8-bit words. Its access time determines the device's speed rating. The 32K-byte EPROM occupies half of the program memory (or EPROM) plane. The EPROM troller architectures.

Dual or Single Memory Planes

8051-family microcontrollers have two external memory planes — program and data. 8096-, 80188-, and 68xx-family microcontrollers have only one program/data plane. The 87C75PF's user-configurable double or single-plane modes work with any 8-bit microcontroller architecture.

Relocatable SFRs

The 87C75PF has five special function registers (SFRs):

- Port 1 latch
- Port 2 latch
- Port 1 Pin
- Port 2 pin
- Plane select.

Port-latch registers allow the microcontroller to change port-pin output levels. The microcontroller can read the port latches to recall the last value written. A microcontroller can determine external pin levels by reading the port-pin locations.

During programming, the plane select register determines whether the EPROM array or the configuration registers are being programmed. More special function register details are described later in this application note.

Device Reconfiguration

Non-volatile (EPROM cell) device-configuration registers configure the 87C75PF for microcontroller compatibility. Configuration registers can be programmed to:

- relocate the EPROM array in the memory map
- relocate the SFRs in the memory map
- combine the EPROM and SFR planes
- change the reset pin's active polarity
- insert transistor pull-ups on port pins.

SYSTEM PERFORMANCE

Every system component influences performance. Performance encompasses speed, system noise, and power consumption. A typical expanded-mode controller application uses many chips to increase memory and recover lost I/O. Figure 3 shows an improved, but more expensive, alternative to the system in Figure 2. "Glue" chips between the controller and peripherals delay address signals. To optimize system speed, fast, expensive glue chips, memory, and peripheral devices are required.

Multiple chip solutions consume significant power and inject noise into a system. A beefed-up, well regulated power supply will relieve symptoms, but adds significantly to cost, board size, and weight.

THE 87C75PF SOLUTION

Figure 4 shows the same system using the 87C75PF — a two chip solution!

The 87C75PF furnishes a no-glue interface to 8051-based systems and all other Intel-architecture embedded controllers. The Port Expander's flexible, user-programmable memory map and alterable control signals simplify 8051, 8096, and 80188 connections.

Examples in this application note show how the 87C75PF works with various microcontrollers. An 8051/87C75PF system that takes advantage of high-level compiled languages and an in-system programmable example will also be shown.

SYSTEM INTEGRATION

Intuitively we all recognize the benefit of system integration — chip-count is reduced.

Just as important are:

- small board size with few layers
- increased performance
- decreased design time
- optimized software devolopment
- reduced inventory
- less incoming inspection
- increased system reliability
- simplified manufacturing.

Cost is a prime consideration. The itemized cost of discrete components is only one parameter. Until the benefits listed above are quantified, realistic system costs can't be determined. Hardware design and software development time are significant up-front expenses. Multiple-chip systems incur substantial

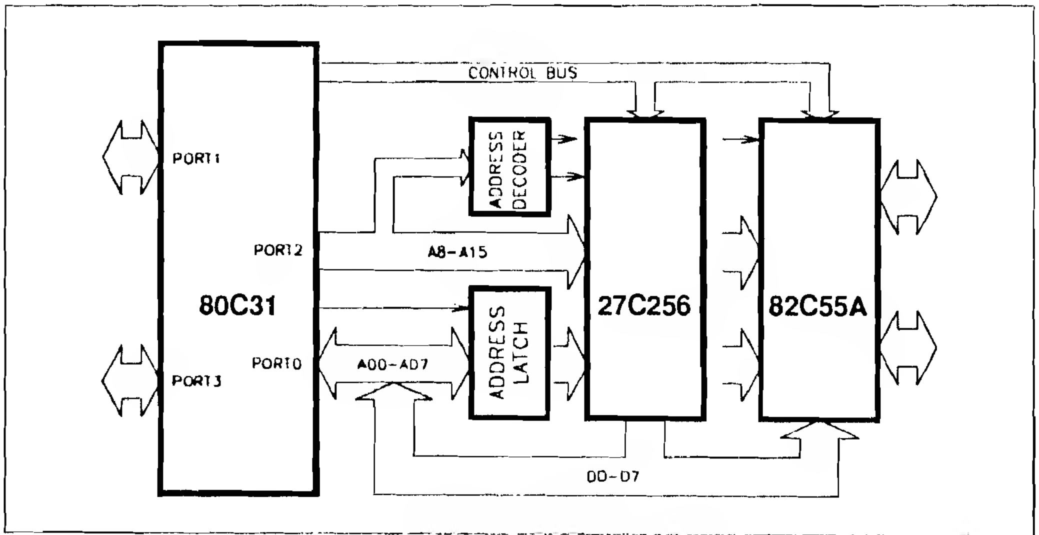

Figure 3. A simplified multiple-chip system.

In its default configuration, the 87C75PF is compatible with the 8051's two-plane architecture. It is easily reconfigured for single-plane 8096 architecture. Remapping the memory planes makes the device compatible with 80188 and 68xx architectures.

Various microcontrollers have different reset input levels. The 8051's reset is active-high while the 8096's is active-low. The 80188 has an active-low reset input and active-high synchronous reset output. The Port Expander's configurable reset polarity can work with active-high or active-low microcontrollers.

If the I/O ports are used only as outputs, a "push pull" drive is desirable. Port 1 and/or port 2 can be configured to have active pull up transistors rather than open drain or quasi-bi-directional outputs.

"No-glue" Microcontroller Interface

The 87C75PF's internal address latches, address decoders, reconfigurable memory planes, and alterable control inputs allow no-glue interfacing to any Intel microcontroller. The 87C75PF makes expanded-mode, two-chip microcontroller systems a reality.

Quick-Pulse Programming

Intel's microcontroller, peripheral, and EPROM products employ the industry's fastest, most reliable Quick-Pulse Programming™ algorithm. Optimized Quick-Pulse Programming equipment can program the 87C75PF in four seconds.

In-circuit Programming

With its integrated features, the 87C75PF is easily programmed in-system. Built-in address latches, address decoders, and flexible control inputs enable the system's microcontroller to program the Port Expander. The section "80C51 In-system programming" describes this technique.

Packaging

For systems requiring periodic reprogramming, prototyping, or hermetic packages, the 87C75PF is available in a 40-pin ceramic DIP (CERDIP) package. PLCC packaging is available to further reduce board size and provide for surface mount and automated manufacturing.

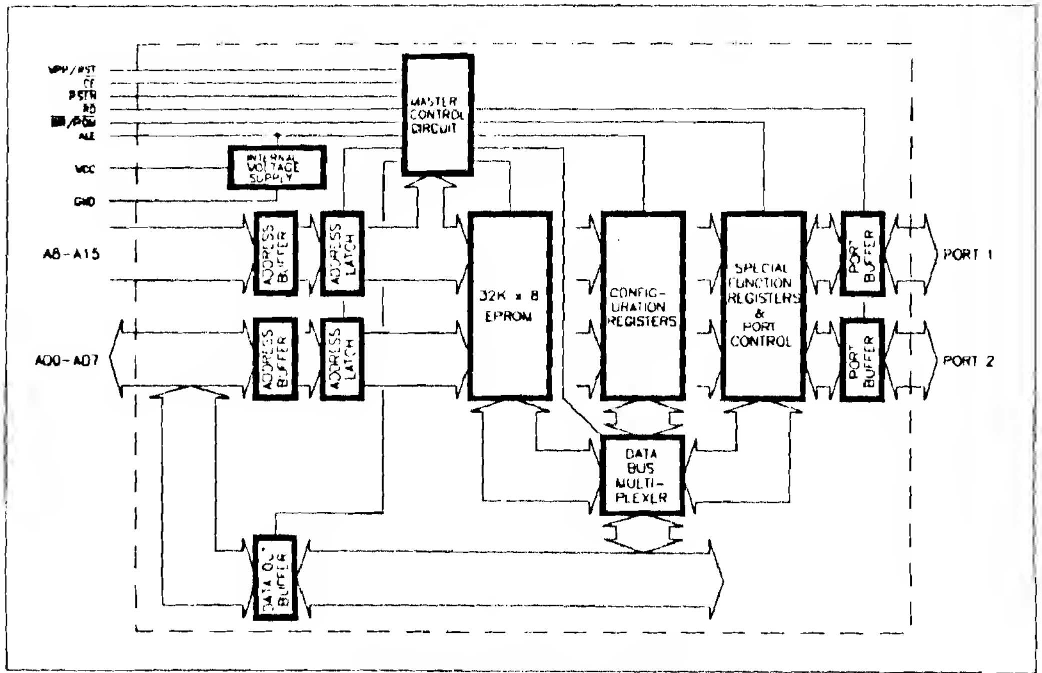

Figure 5. 87C75PF Block Diagram.

87C75PF FUNCTIONAL BLOCKS

planes EPROM, special function registers, and configuration registers.

The block diagram shows device inputs on the left and outputs on the right. Sixteen address lines enter the device and their states are latched by ALE. The lower eight address pins are multiplexed with data. $\overline{PSEN}$ (Program-Store ENable) gates the device's EPROM data. $\overline{RD}$ gates SFR data. $\overline{WR}/\overline{PGM}$ controls SFR data writes. $\overline{CE}$ is the master chip enable input. V_{pp} (the programming voltage input) is multiplexed with RST (reset). V_{pp} is required only during programming. Asserting RST sets port latches to "1s" during operating mode.

Port 1 is an 8-bit open-drain port with optional "CMOS" drive capability. Port 2 has 8 quasi-bi-directional pins, also with optional "CMOS" drive.

DEVICE PINOUTS

similar to Intel's 27210 megabit EPROM. The device's pinouts are compatible with most programming equipment capable of programming 27210 EPROMs.

Figure 6 shows the CERDIP pinout. The left side has sequential address and data inputs. The ground pin (GND) separates lower and upper address lines for better noise immunity. Ports are logically placed on the device's right side. Port 1, which is open-drain, is near V_{cc}. SIP pack resistor pull-ups added externally to port 1 have easy access to V_{cc}.

Figure 7 shows the PLCC pinout. PLCC leads are in the same sequence as the CERDIP pinout. No-connect (NC) and don't-use (DU) leads are inserted at strategic locations. Future enhancements will use these leads for expanded features. DU leads should be left unconnected.

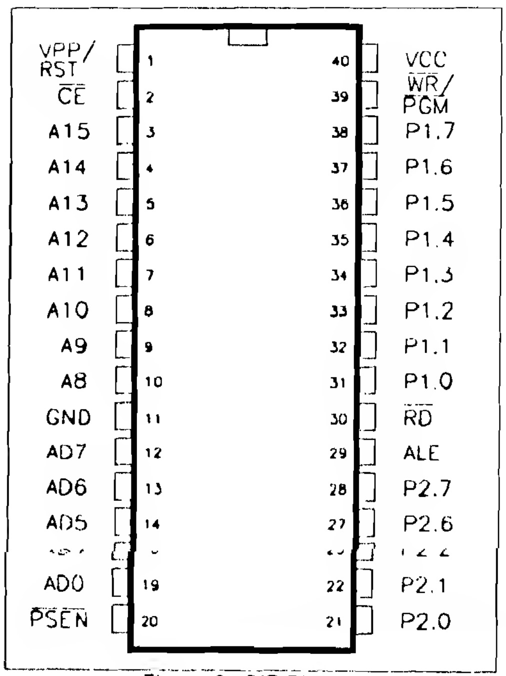

Figure 6. DIP Pinout.

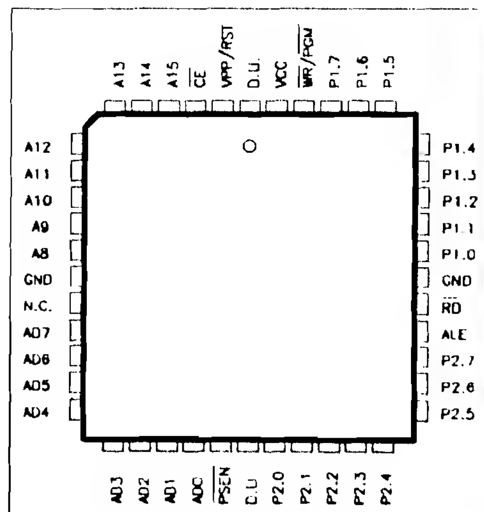

Figure 7. PLCC Pinout.

3 PLANE MEMORY MAP

The 87C75PF has three memory planes: EPROM, SFR, and configuration. Two planes, EPROM and SFR, are available during operating mode. The configuration plane is present under special programming conditions. Figure 8 shows the three memory planes, conditions when they are present, control signals that access them, and memory locations they occupy.

EPROM Plane

The 32K-byte EPROM fills the lower half (0000h-7FFFh default) of the 64K-byte EPROM plane. This conforms to 8051- and 8096-family microcontrollers that have reset and interrupt addresses in the bottom half of the memory map. The EPROM array can adapt to 80188- and 68xx family microcontrollers by moving it to high memory (8000h-FFFFh). $\overline{PSEN}$ is the EPROM array's operating- and programming-mode read control. $\overline{WR}/\overline{PGM}$ strobes data into the array only during programming mode.

SFR Plane

Special function registers are located in the SFR plane. They occupy low-addresses in a relocatable 2K-byte block (default addresses F800h-FFFFh). The 2K SFR block can be placed on any 2K-byte address boundary to match microcontroller architecture requirements. $\overline{RD}$ and $\overline{WR}/\overline{PGM}$ control reads and writes to/from this plane.

Configuration Plane

The configuration plane contains non-volatile EPROM registers that determine the device's configuration. This plane is available only when high voltages are applied to special pins. PROM programming equipment can use this plane to identify the device, read its present configuration, and program new configurations. Memory-mapped registers can be programmed to:

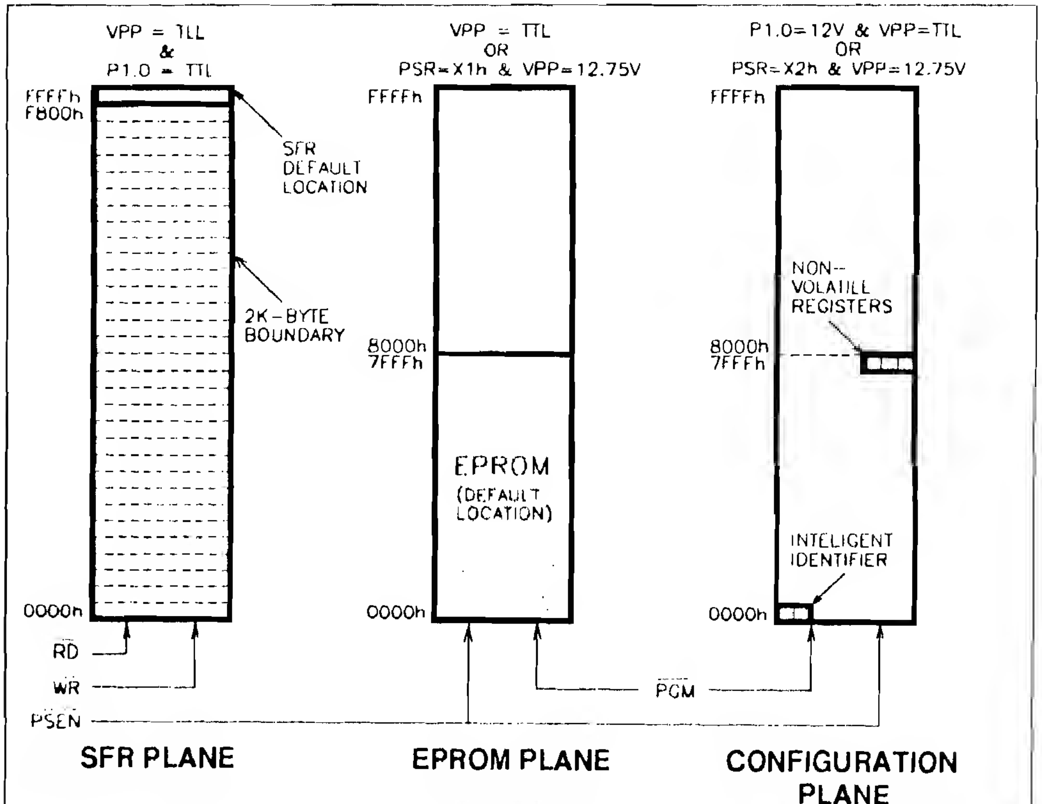

Figure 8. The 87C75PF has three internal memory planes -- SFR, EPROM, and Configuration.

- move the EPROM array
- move the SFR block
- combine the EPROM and SFR planes
- combine $\overline{PSEN}$ and $\overline{RD}$
- change RST's polarity
- insert pull-up transistors on port output drivers.

Device reconfiguration will be covered further in the "Architecture compatibility" section.

Plane Select Register

The plane select register (PSR) occupies address F810h in the SFR plane (Figure 9). This register's value determines which plane, EPROM or configuration, is in programming mode. The following plane is programmed when V_{pp} is raised to its programming voltage if PSR contains:

- xxxxxx00 = programming prohibited
- xxxxxx01 = EPROM plane
- xxxxxx10 = configuration plane
- xxxxxx11 = programming prohibited.

Note that both PSR bits must toggle to change planes. Spurious programming noise is unlikely to alter both bits simultaneously. This safeguard prevents erroneous programming of the wrong plane.

I/O PORTS

The 87C75PF has two 8-bit, bi-directional I/O ports. Each port has two addresses in the SFR plane — port latch and port pin. The port latch register drives port pins; it's the port output register. Byte-wide data written to it is strobed by $\overline{WR}/\overline{PGM}$'s rising edge. This allows individual register bits to be changed without "glitching" unchanged bits. Port latches can be read to determine previously stored values. Redundant RAM locations that contain port values are not required. Asserting RST sets port latches to "1s".

Each port has a pin register. This input register allows a microcontroller to monitor pin status. Although a port latch register may drive a port pin to "1", an external switch can pull it to "0". A software exclusive-OR of latch and pin values will discover the switch closure.

Figure 9 shows the 2K-byte SFR block (default location shown) containing port addresses. Locations F800h-F807h are reserved for port latch addresses; the 87C75PF uses only two of these addresses. Locations F808h-F80Fh are reserved for port pin addresses; again, the 87C75PF uses only two addresses. Each port latch and port pin register contains eight bits; each corresponding to a port pin. Locations F810h-F81Fh are reserved for SFR registers.

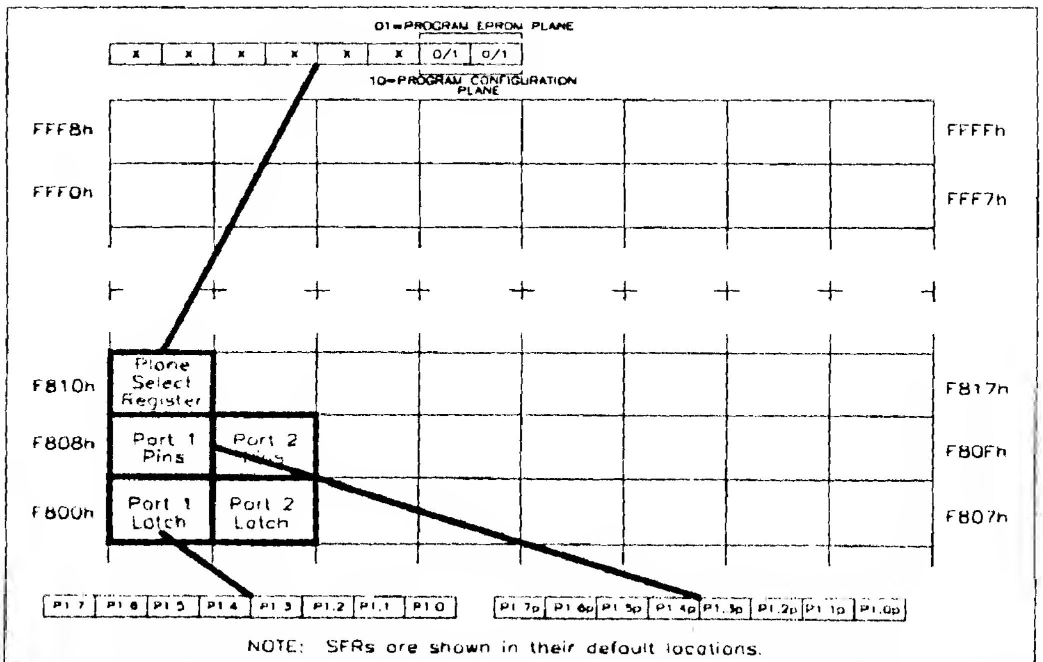

Figure 9. The 2K-byte SFR block contains port latch and pin addresses and Plane Select Register

Port 1

Port 1's default latch address is F800h; its pin address is F808h. Its default configuration is open drain. Other open-drain devices can be "wire-ORed" to port 1 pins. Pull-up resistors can be added externally to provide I_{OH} drive.

Port 1's outputs can be reconfigured to supply CMOS drive. Programming the control level register's P1C bit (CLR.6) inserts active pull-up transistors. This switches port 1 pins faster from V_{OL} to V_{OH} and simplifies interfaces to external CMOS devices. Figure 10 shows port 1's block diagram.

Port 2

Port 2 is similar to port 1. Its latch address is F801h and its pin address is F809h. Its default configuration is quasi-bi-directional. This means that each pin has a weak pull-up resistor. External pull-up resistors can be added to increase the port's I_{OH} drive.

Port 2's outputs can be reconfigured to supply CMOS drive. Programming the control level register's P2C bit (CLR.5) inserts active pull-up transistors. Figure 11 shows port 2's block diagram. Note the difference between port 2's and port 1's output stages. In addition to the weak pull-up resistor, the feedback network senses the pin's V_{OH} level and switches a stronger pull-up resistor into the circuit. A V_{OL} level turns the resistor off. Another addition is the pulsed pull-up. When a port latch value changes from "0" to "1", the CMOS transistor is pulsed to quickly supply current to the pin.

ARCHITECTURE COMPATIBILITY

Every microcontroller family has its own architecture. Each has unique boot-up, interrupt, and vectoring addresses. Some support dual external memory planes while others communicate with only one. External addressing capacity varies from 64K- to 1M-bytes.

The 8051's control signals and software instructions manipulate 5 memory planes. Three planes are inter-

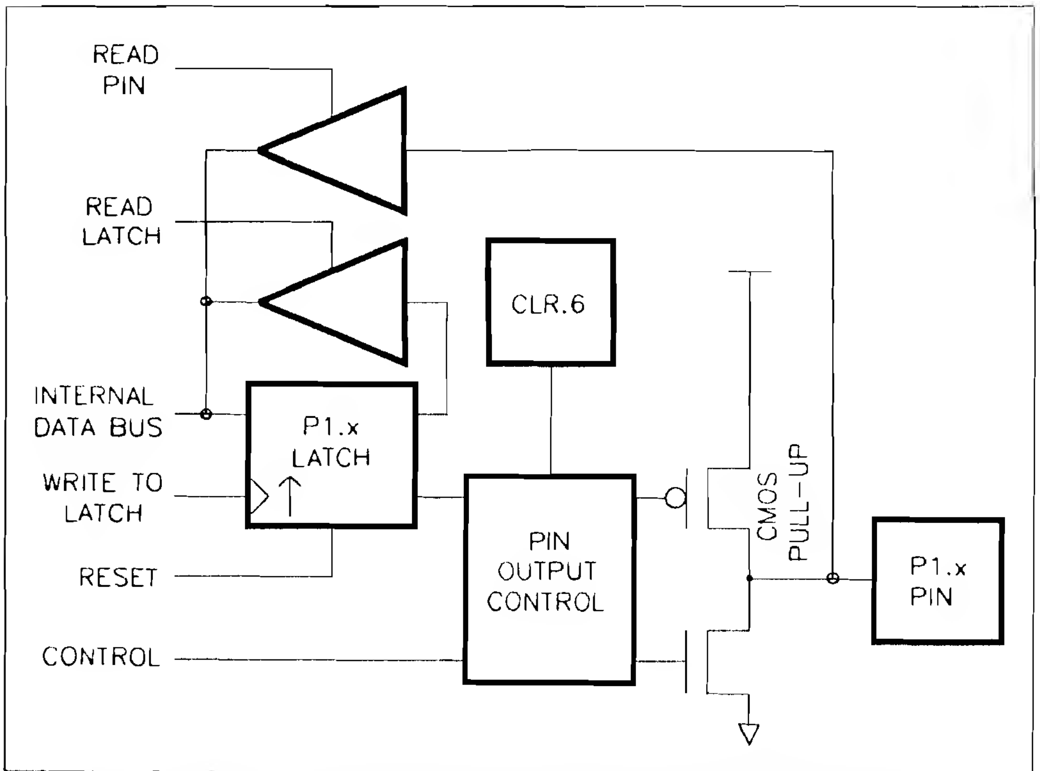

Figure 10. Port 1 is Open-Drain (default) or programmable for active (CMOS) pull-ups.

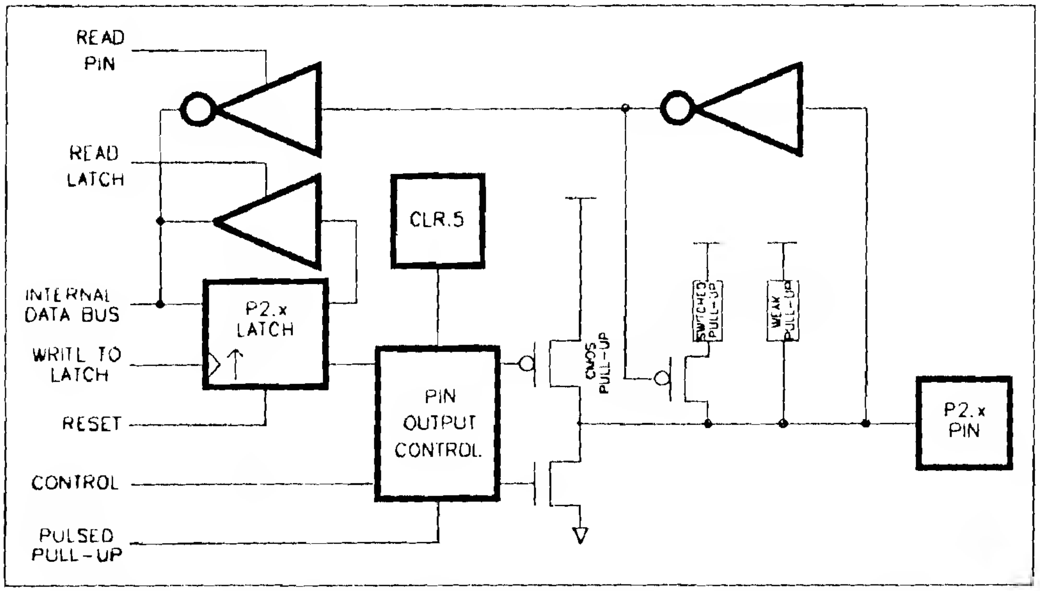

Figure 11. Port 2 is Quasi-bi-directional (default) or programmable for active (CMOS) pull-ups.

nal — on chip ROM/EPROM, RAM/SFR, and bit-addressable registers. Two planes are external — program (EPROM) and data (RAM) memory. The instruction type drives internal and external read, write, and bus signals that select individual planes. 8051 controllers require non-volatile boot-up memory, internal or external, at the bottom of its program memory plane. The 87C75PF's two-plane external-memory architecture (see Figure 12) matches the 8051's architecture. EPROM defaults to the EPROM plane's low-memory and SFRs default to the SFR plane's high-memory.

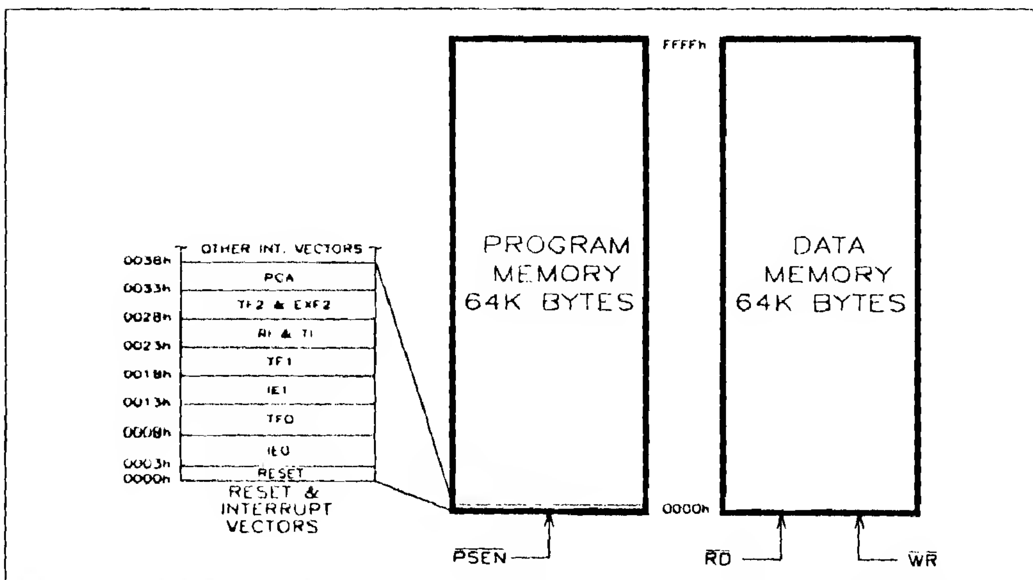

Figure 12. The 8051's two-plane memory has reset and vector addresses in low program-memory.

8096-family controllers are typically used with a single 64K-byte external memory plane (Figure 13). Like the 8051, reset and vector addresses are in low memory. The 87C75PF has an optional single-plane configuration that complements 8096 architecture. The EPROM, located in low memory, is combined with the SFR plane.

Intel's 80188 microprocessor is used primarily in high-end embedded-control applications. Adding ports and memory makes the 80188 one of the most powerful microcontrollers available. The 87C75PF provides much of this hardware in a single package. The 80188 has a single memory plane. Unlike 8051 and 8096 controllers, its boot-up address is at the top of its 1M-byte address space (Figure 14). The 87C75PF can be configured for a no-glue 80188 interface.

The 87C75PF's flexibility simplifies hardware interfacing with many other microcontrollers. A 68xx controller, for example, has boot up vectors at the top of its 64K-byte single-plane memory space. The Port Expander's memory map can be configured, much like that used by the 80188 (Figure 14), to accommodate 68xx controllers.

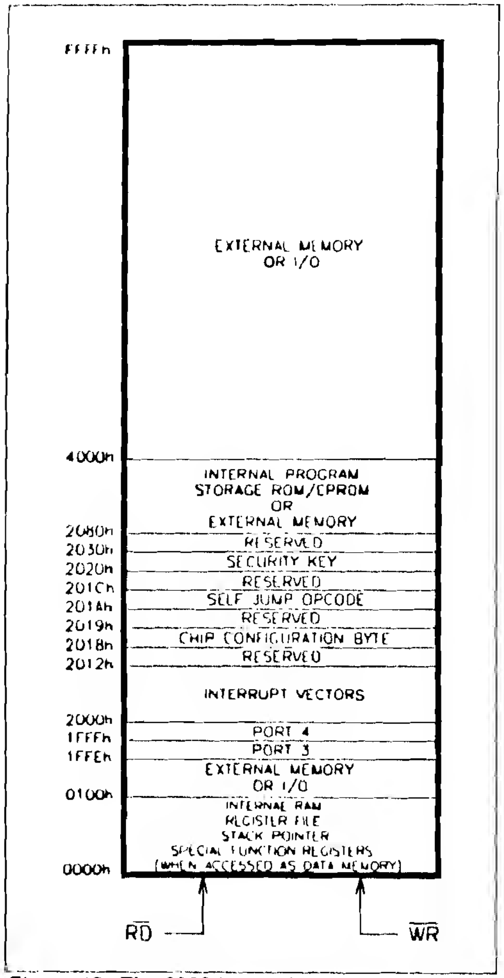

Figure 13. The 8096 has a single memory plane.

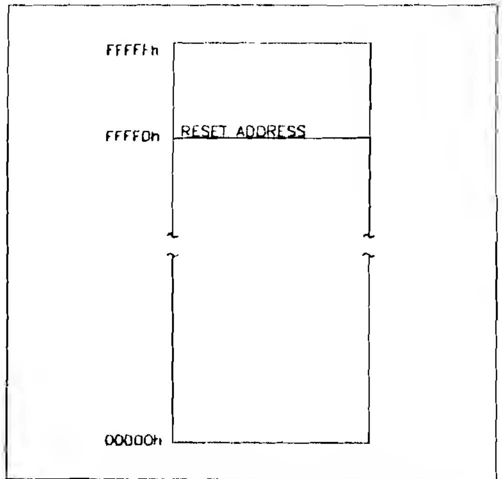

Figure 14. The 80188 boots up at the top of its 1M-byte address space.

Default Configuration

Ultraviolet light exposure will erase the 87C75PF's EPROM array and non-volatile configuration registers. The EPROM, SFRs, and other user-configurable options default to:

- two memory planes — EPROM and SFR
- EPROM at 0000h-7FFFh
- SFR block at F800h-FFFFh
- reset (RST) active-high
- port 1 open drain
- port 2 quasi-bi-directional.

Changing the Reset Polarity

8051-family microcontrollers have active high reset inputs. 8096, 68xx, 80188, and special 8051-architecture controllers have active-low resets. The 80188 also has an active-high synchronous reset output.

The Port Expander's alterable reset input (RST) can match any microcontroller. When erased, the 87C75PF's RST is active high. Programming the configuration plane's control level register bit CLR.7 changes RST to active-low (see Figure 15).

Changing Port Output Drive

If port 1 and/or port 2 are used only as outputs, it may be preferable to have CMOS-type output levels. Programming CLR.6, P1C, and/or CLR.5, P2C (see Figure 15), inserts active pull-up transistors in port output buffers. These transistors supply higher current and faster switching than open drain or quasi-bi-directional outputs.

Moving the EPROM

The 87C75PF's EPROM can be relocated to the upper half of its 64K-byte memory map. When erased, the EPROM is correctly positioned in low memory for 8051- and 8096-family controllers. Programming the configuration plane's EPROM Location bit, ELR.7 (Figure 16), moves the EPROM to high memory for 80188 and 68xx compatibility.

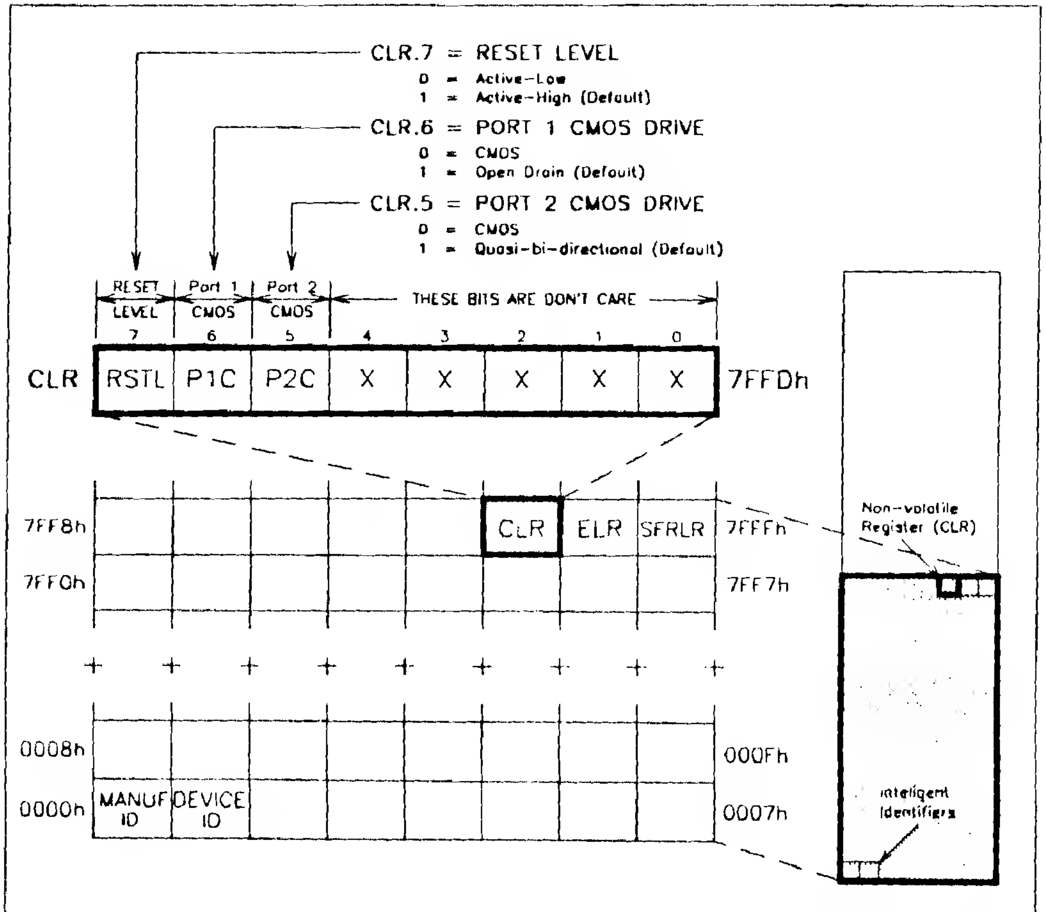

Figure 15. The Control Level Register (CLR) determines the reset pin's polarity and CMOS port drive.

Double- and Single-plane Configurations

The 87C75PF has two operating-mode memory-planes — EPROM and SFR. These planes share identical memory addresses. The EPROM plane is selected when $\overline{PSEN}$ is TTL-low. The SFR plane is selected when either $\overline{RD}$ or $\overline{WR}$ is TTL-low. 8051 microcontrollers use $\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}$ to select two external memory planes. 8096 controllers have only $\overline{RD}$ and $\overline{WR}$, some versions have an "INST" output that allows external circuitry to determine when instructions are being issued. Most other microcontrollers provide read and write signals that control only one memory plane.

Programming the 87C75PF's overlap bit, OVLP (ELR 6), converts the device from dual-plane to single-plane (see Figure 16). When ELR.6 = "0", $\overline{PSEN}$ and $\overline{RD}$ are internally combined. Both memory planes are active if either is TTL-low.

8051 applications that use code compiled from high-level languages find this especially useful. Some high-level languages can't distinguish between data-plane and program-plane addresses. For example, look-up tables stored in the same EPROM as program instructions require $\overline{PSEN}$ to be asserted. However, a compiler interprets look-up table instructions as data fetches. It assigns code that asserts $\overline{RD}$ instead of $\overline{PSEN}$. A typical hardware solution uses an AND gate to combine $\overline{PSEN}$ and $\overline{RD}$. This forms one memory plane that is accessed by either signal. Programming the 87C75PF's OVLP bit provides this "AND" function.

This bit also permits the SFRs to overlap the EPROM array. This allows multiple Port Expanders to be used in single-plane applications. For example, two Port Expanders can be used in an 8096 system (see Figure 22). Normally, two 87C75PF's 64K EPROM bytes consume the entire address space leaving no room for port addresses or external RAM. When ELR 6 = "0" and the device's 2K byte SFR block overlaps its EPROM array, 2K EPROM bytes are sacrificed to make room for the SFRs and external RAM. Under these conditions, the 87C75PF remains in a high impedance state during any access to the 2K-byte SFR block except for the five valid SFR addresses (see Figure 9).

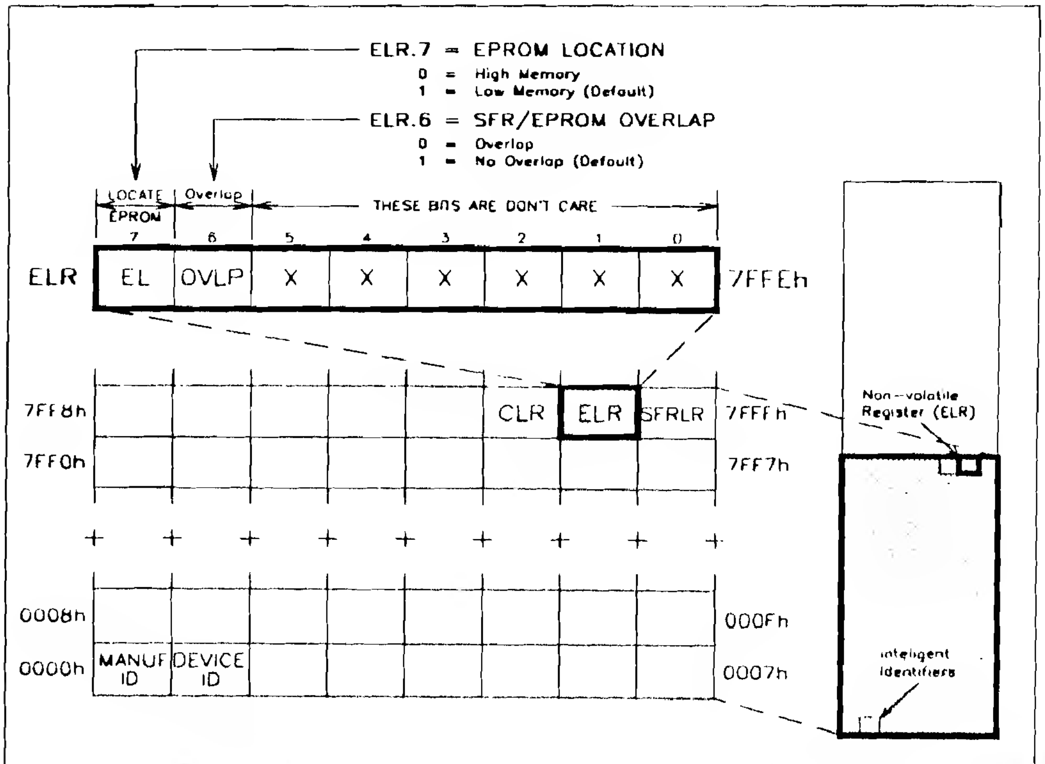

Figure 16. The EPROM Location Register determines the EPROM's memory-map location.

Moving the SFR Block

The 2K-byte SFR block's default location is F800h-FFFFh in the SFR plane. This location is fine for 8051 and 8096 applications. However, 80188 and 68xx-family controllers have boot-up and vector addresses in this address range, EPROM should be located here.

The SFR block can be moved to any 2K-byte device-address boundary. The SFR location register's (SFRLR) five bits determine the SFR-block's most-significant address bits. When erased, these bits are all "1s", placing the SFRs at 11111xxx xxxxxxxxb or F800h-FFFFh. Programming the SFRLR to 01111xxx, for example, relocates the SFR-block to 7800h-7FFFh (just below the EPROM array when it's at the top of memory, 8000h-FFFFh). Programming SFRLR to 00000xxx moves the SFRs to the bottom of memory, 0000h-07FFh. Figure 17 shows the SFRLR and its bit definitions.

Programming the Configuration Plane

The 87C75PF data sheet describes detailed programming requirements. PROM programming equipment makes device reconfiguration easy. Down-loading EPROM code (from 0000h to 7FFFh) to the programmer is the same as for any 256K PROM device. The programmer allows editing of CLR, ELR, and SFRLR codes to reconfigure the device. Once programming commences, the EPROM array and the configuration registers are programmed automatically.

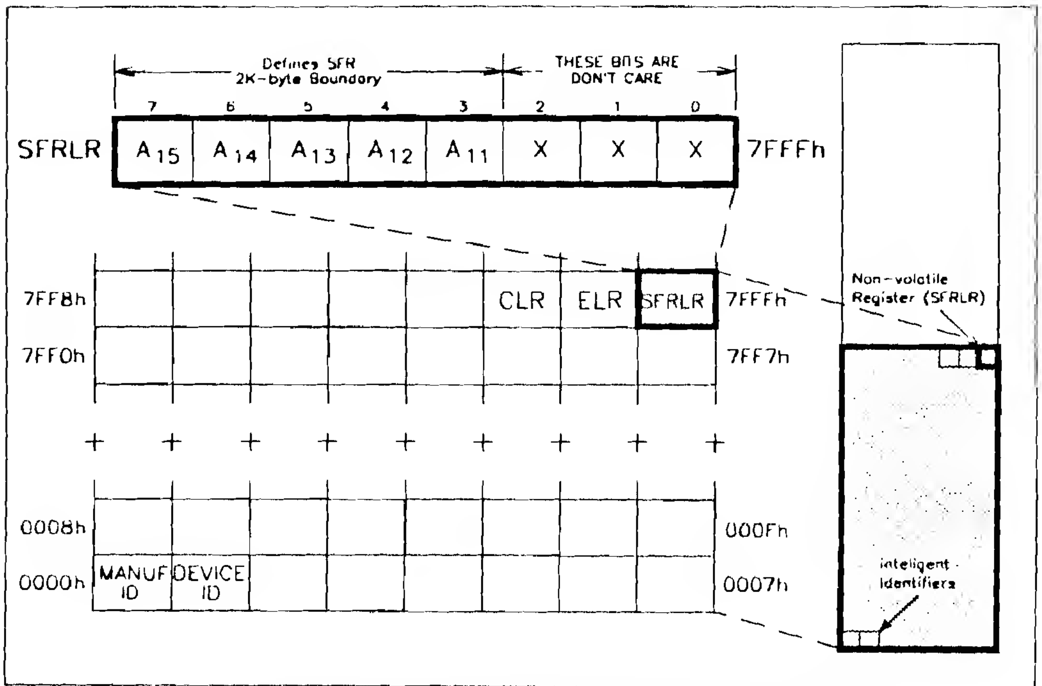

Figure 17. The SFRLR determines the 2K-byte SFR block's base address.

87C75PF APPLICATIONS

Now that you're familiar with how the Port Expander is organized and reconfigured, this section highlights some application examples. You'll see how the 87C75PF connects to 8051, 8096, 80188, and 68xx microcontrollers. Also shown are more sophisticated applications that use multiple Port Expanders and one that allows the microcontroller to program its own Port Expander. All of the applications illustrated show microcontroller/Port Expander interfaces, memory maps, and configuration register (CLR, ELR, SFRLR) values.

80C31 + 87C75PF

8051-family controllers usually operate in two-plane mode. To use external program memory (EPROM) exclusively, the controller's external access pin, $\overline{EA}$, is tied to ground. Port 2 supplies upper addresses, A_8-A_{15}. Port 0 becomes the multiplexed lower-address/data bus, AD_0-AD_7. $\overline{PSEN}$ is the program memory read strobe. $\overline{WR}$ and $\overline{RD}$ (port pins P3.6 and P3.7) control external RAM and other read/write devices. RST is active-high on most 8051-family microcontrollers. Some special-purpose '51-based controllers have active-low resets.

Figure 18 shows a typical 80C31 + 87C75PF no-glue application. The 87C75PF's EPROM, SFR, and control-signal default-settings are already configured. Programming the large XX place holders shown in the CLR register enables CMOS port drive.

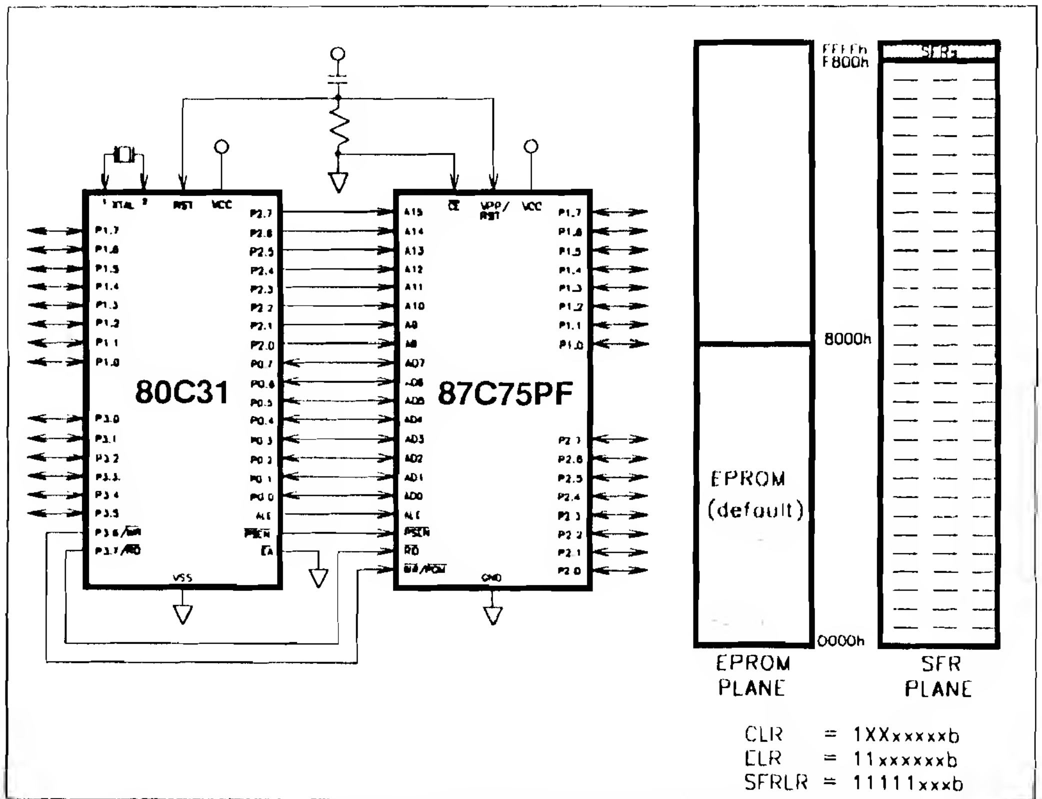

Figure 18. The 87C75PF's no-glue interface takes advantage of the 80C31's two-plane memory map.

80C31 + Two 87C75PFs

High-end applications, such as telecommunications, require sizable program memories and numerous I/O ports. Many of these applications use 8051-family microcontrollers. Two 87C75PF Port Expanders supply added I/O while furnishing EPROM — without using "glue" devices!

Figure 19 shows two Port Expanders in an 80C31 system. Port Expander 1's EPROM is in its default low-memory location (0000h-7FFFh). Its SFR block is moved to F000h, out of Port Expander 2's SFR range (F800h). Port Expander 2's EPROM is moved to high-memory (8000h-FFFFh). Each device's configuration register values are shown below the memory map. This configuration provides 16 additional I/O pins, 64K EPROM bytes, and leaves 60K for RAM and other memory-mapped devices.

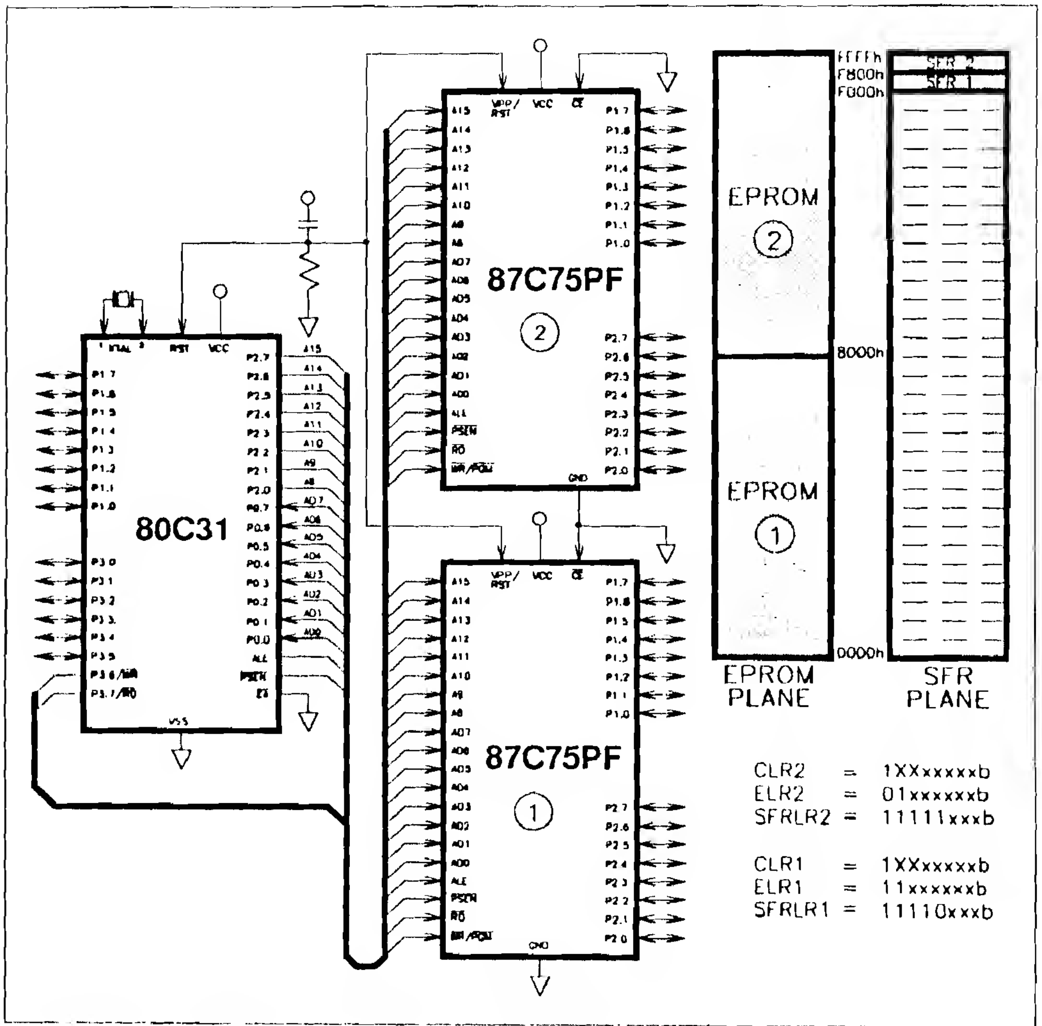

Figure 19. Two 87C75PFs provide 16 I/O pins, 64K EPROM bytes, and room for 60K of RAM.

High-level Language 80C31 + 87C75PF

The 8051's two plane flexibility challenges hardware and software engineers' creativity. Its two planes logically separate program and data planes to create 128K bytes of memory in a 64K address space. However, many applications have look-up tables in non volatile memory, usually in the same EPROM that contains program code. Unique assembly-language instructions drive hardware signals, $\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}$, to determine which plane is active.

Some compiled, high-level programming languages, however, have a hard time dealing with two-plane memories. They can't determine which 8051 instruction to use when look up tables occupy the program plane. They usually assign an instruction that activates $\overline{RD}$, rather than $\overline{PSEN}$.

The typical solution forces the system to operate in single-plane mode by combining $\overline{PSEN}$ and $\overline{RD}$ with an AND gate. If either signal is TTL-low, the AND gate's output drives a common external-memory read signal. A compiler can now assign its typical "read from data memory" instruction.

The Port Expander has this "AND" function built in. Programming the configuration plane's Overlap bit, ELR.6, internally combines $\overline{PSEN}$ and $\overline{RD}$; if either is at TTL-low EPROM or SFR data, depending on the address, is read. Figure 20 shows a typical high-level-language application.

Programming this bit also allows the SFR-block to overlap the EPROM in single-plane applications. If, and only if, these blocks overlap, 2K EPROM bytes are sacrificed to make room for the SFR block. The "8096 + two 87C75PFs" section illustrates this.

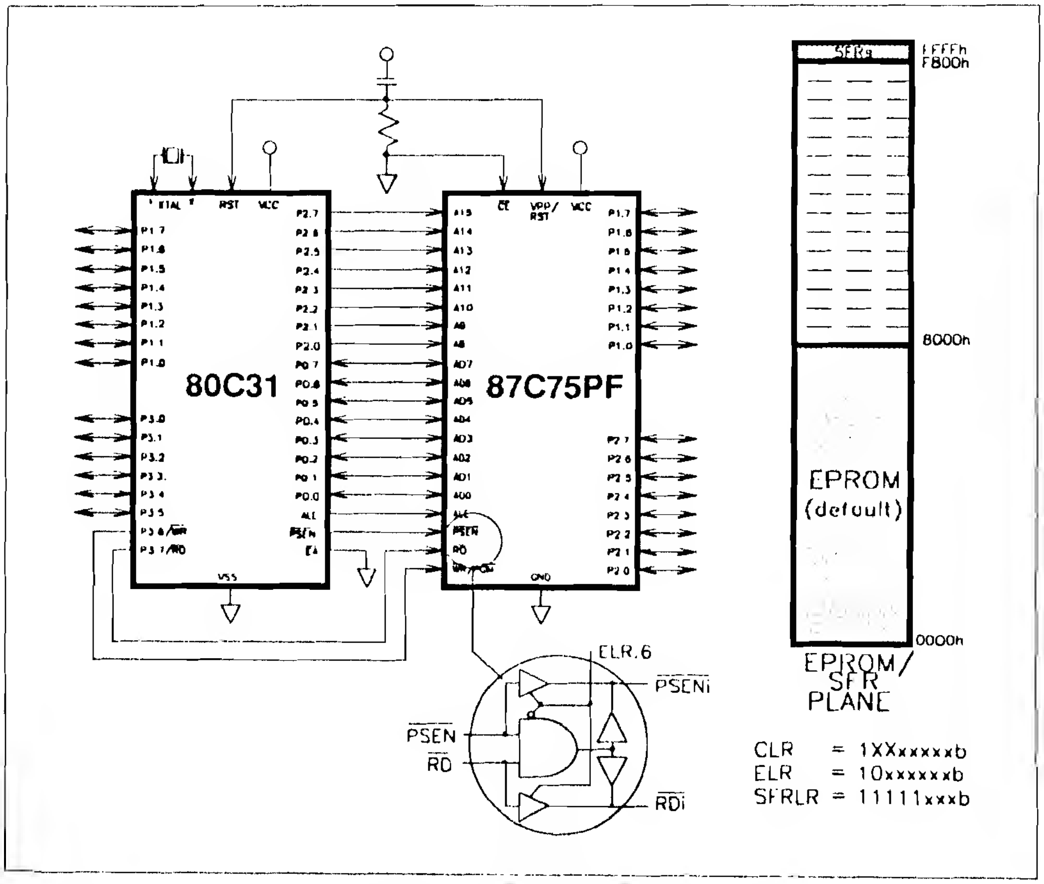

Figure 20. Programming ELR.6 combines $\overline{PSEN}$ and $\overline{RD}$ to form a single memory plane.

8096 + 87C75PF

8096-family 16-bit microcontrollers can also operate in 8-bit mode. These high performance controllers manage applications that are I/O intensive and, as a result, require large EPROM arrays. The 87C75PF expands the I/O while providing the EPROM.

The 8096 accesses a 64K-byte single-plane memory. Its memory map is similar to the 8051's. External EPROM is required at its low-memory boot-up location (2080h). The 87C75PF's EPROM and SFRs are appropriately located.

The 8096's reset input ($\overline{RES}$) is active-low. Programming the Port Expander's reset level configuration bit, RSTL (CLR.7), makes RST's polarity active-low.

The 87C75PF is converted to single-plane mode by either tying $\overline{PSEN}$ and $\overline{RD}$ to the 8096's $\overline{RD}$ pin or by programming ELR.6, the overlap bit. If the latter option is chosen, the unused input, $\overline{PSEN}$ or $\overline{RD}$, should be tied to V_{CC}. Figure 21 shows a "no-glue" 8096 + 87C75PF application.

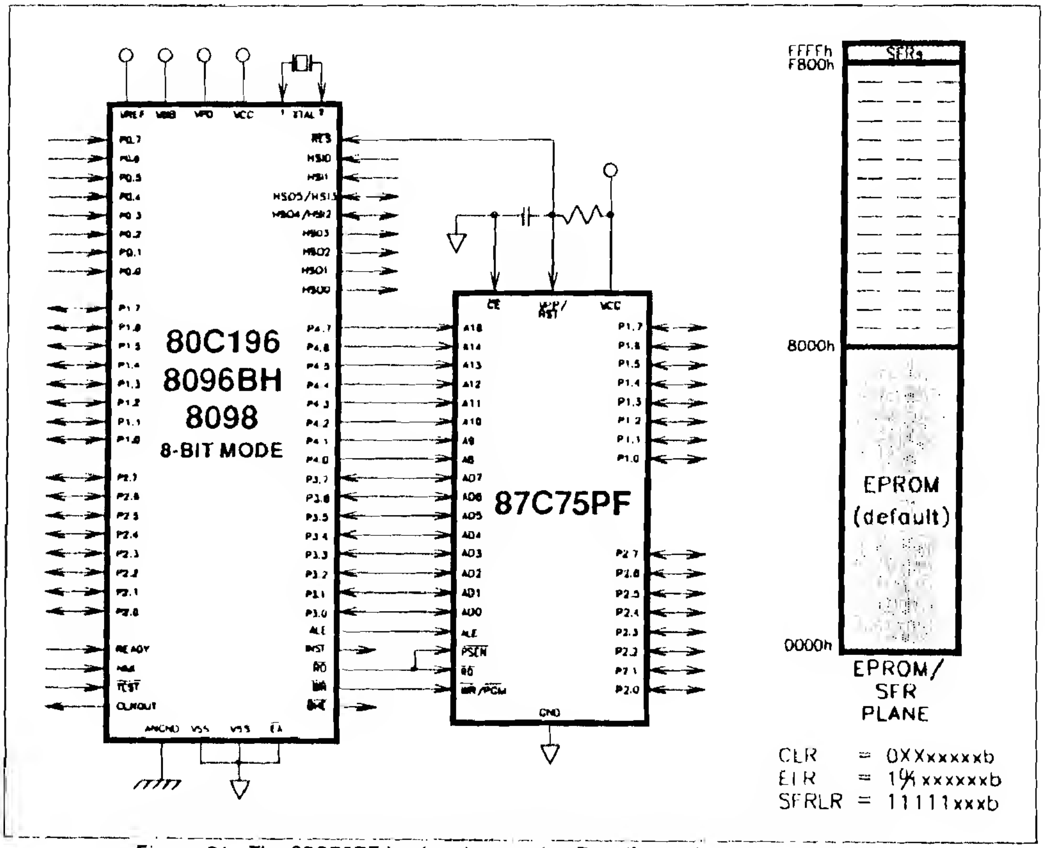

Figure 21. The 87C75PF is also the no-glue Port Expander for 8096 systems.

8096 + Two 87C75PFs

Single plane 8096 applications can use two Port Expanders. Figure 22 shows this no-glue, three-chip system.

Port Expander 1 has its EPROM in default low-memory. Its SFR block is mapped over its EPROM; location 7800h is arbitrarily chosen. Programming 01111xxxb into SFRLR moves the SFR block. Programming ELR 6 (to "0") overlaps the EPROM and SFR planes; one plane is formed. This bit also tells the Port Expander that its SFRs are intentionally mapped over its EPROM. The device sacrifices 2K EPROM bytes to make room for the SFR block. Any access to this 2K-byte block, except valid port and PSR addresses, places the external data bus in a high impedance state. External RAM can occupy the 2K-byte space.

Port Expander 2 is also reconfigured. Its EPROM is moved to high memory by programming ELR 7. Its SFR block must overlap its EPROM array; 8000h is arbitrarily chosen. Port Expander 2's overlap bit, ELR 6, is programmed to form a single plane and to tell the device that its SFRs are intentionally mapped over its EPROM, like Port Expander 1. This configuration supplies four additional 8-bit ports, 60K EPROM bytes, and still leaves 4K bytes free for RAM.

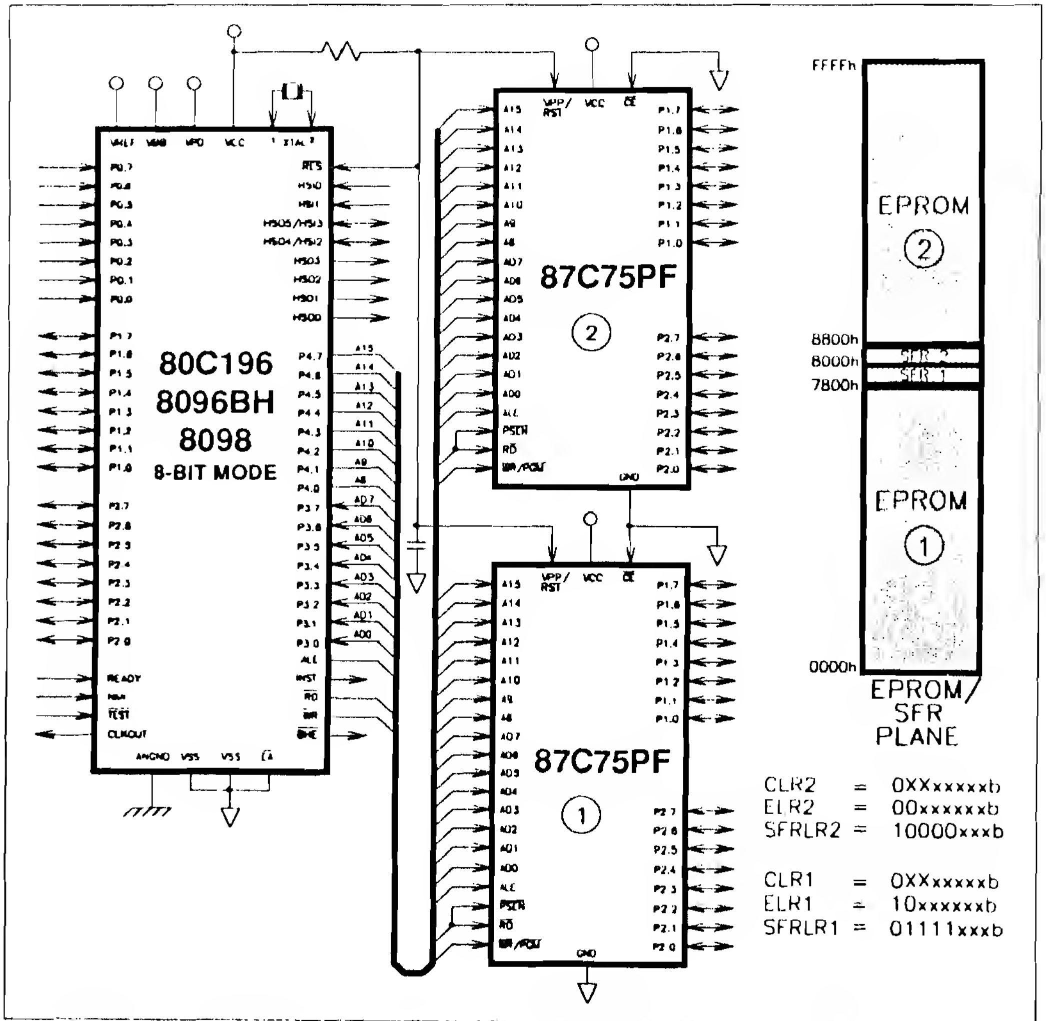

Figure 22. Two 87C75PFs add 16 I/O pins, 60K EPROM bytes, and leave room for 4K of RAM.

80C188 + 87C75PF

The 80C188 found its niche in high-end embedded control applications. This CPU, when combined with RAM and the Port Expander, becomes a powerful embedded controller. Its 1M-byte address range accommodates several Port Expanders and large amounts of RAM. Although the 80C188 has two planes, memory and I/O, the Port Expander works best in the memory plane. Figure 23 shows a simple 80C188 + 87C75PF system.

The 80C188 boots up at address FFFF0h. The 87C75PF's EPROM array is moved to its high memory (8000h FFFFh) by programming ELR.7. The SFR block must be moved to lower memory outside of EPROM-block addresses, (F7800h is shown). Programming the overlap bit, ELR.6, or tying $\overline{PSEN}$ and $\overline{RD}$ to the 80C188's $\overline{RD}$ combines the EPROM and SFR planes. The processor's $\overline{UCS}$, connected to the 87C75PF's $\overline{CE}$, selects the Port Expander in the upper address range. The 80C188's reset input, $\overline{RES}$, is active low. Programming the 87C75PF's RSTL bit, CLR.7, converts RST to active-low. the 80C188 also has an active-high synchronous reset output. This output can be connected to the 87C75PF's RST without reconfiguring RST's polarity.

80C188 systems usually have larger RAM arrays than typical microcontroller applications. Figure 23 shows the simple RAM interface. The RAM does not contain its own address latches, so an 8-bit latch must be used to capture addresses A_0-A_7.

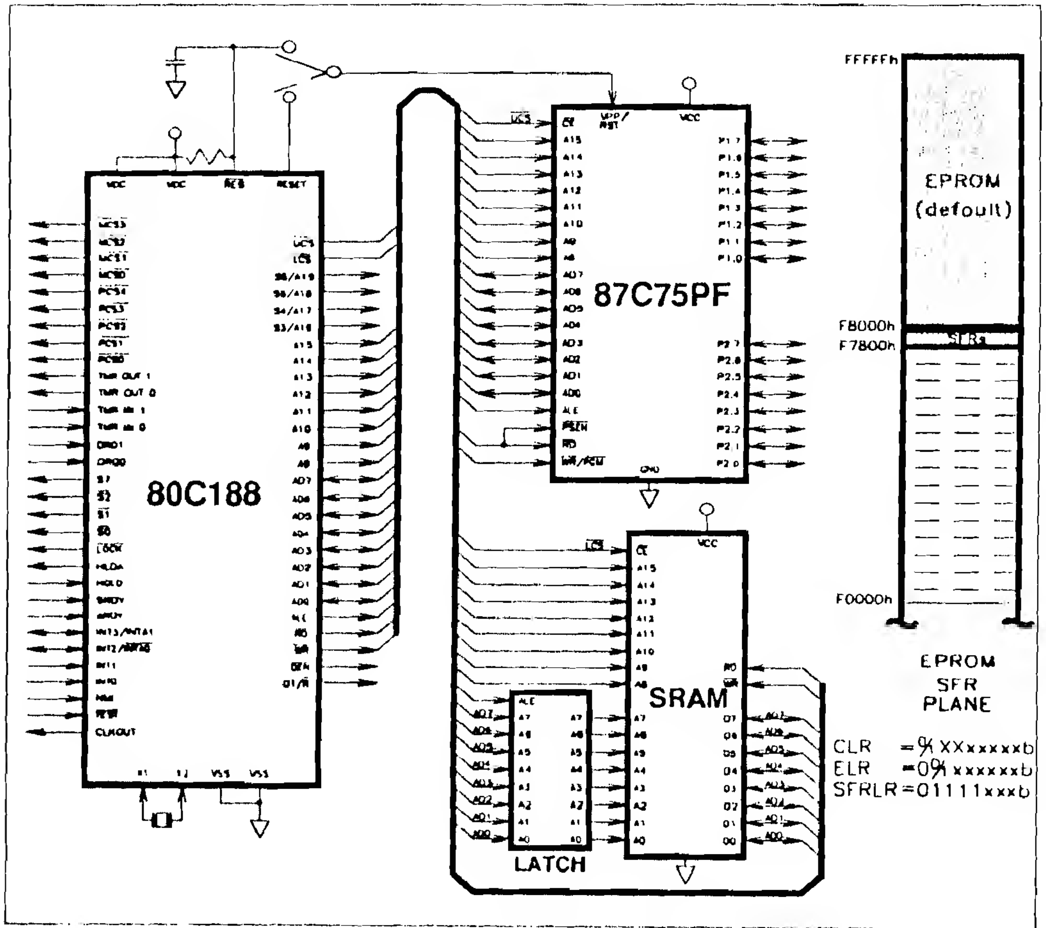

Figure 23. The Port Expander and SRAM make the 80C188 a powerful embedded controller.

68xx + 87C75PF

The microcomputer industry's peripheral- and memory interface standard dictates chip enable, output enable, and write-enable polarities. All are active-low. The 87C75PF conforms to this industry standard.

Like Intel controllers, 68xx-family microcontrollers use multiplexed address/data pins. However, they differ in two significant ways. First, 68xx controllers have high-memory reset- and interrupt-vector addresses. Address A_{15} is logic-high during vector accesses. Second, read and write controls are functions of $R/\overline{W}$ and E (clock output). Combinational logic must convert $R/\overline{W}$ and E to industry standard $\overline{RD}$ and $\overline{WR}$ signals.

The 87C75PF's memory map can be reconfigured and its two memory planes combined to simplify 68xx interfaces. Its RST polarity can match a 68xx's active-low reset. All that's required to complete the interface is to condition $R/\overline{W}$ and E to $\overline{RD}$ and $\overline{WR}$. Figure 24 shows a 68xx + 87C75PF system and its memory map.

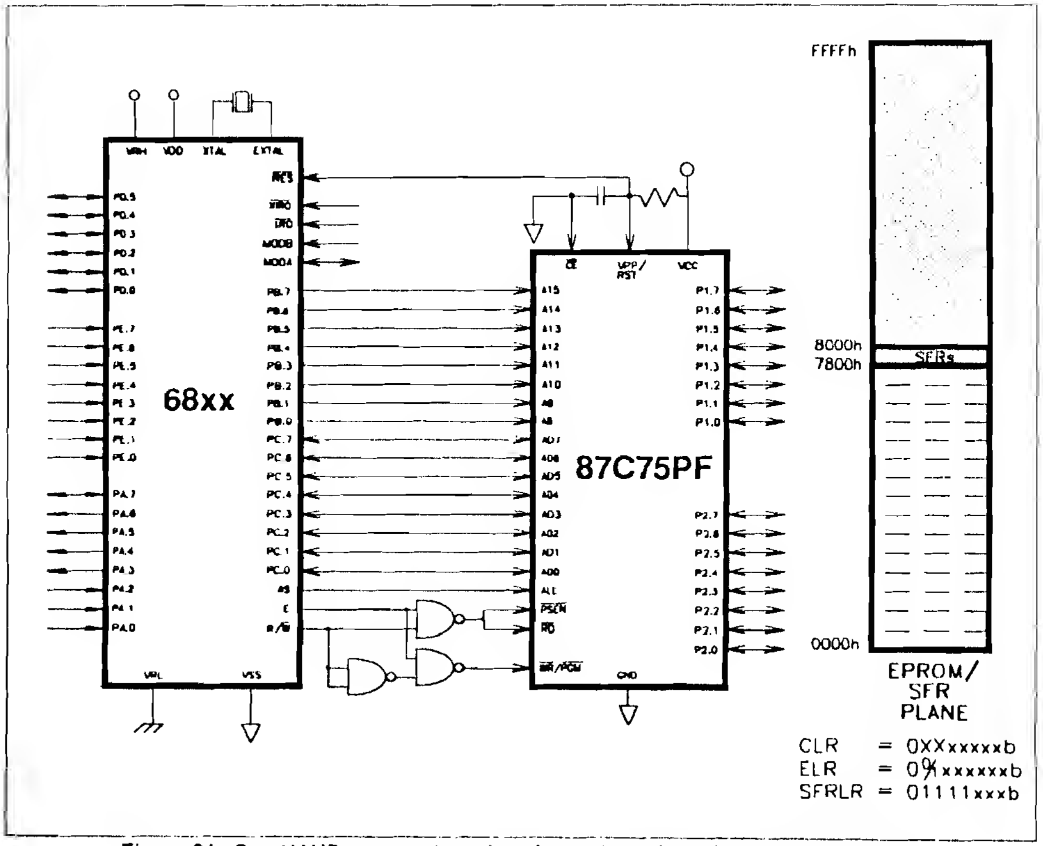

Figure 24. One NAND-gate package interfaces the 87C75PF to 68xx controllers.

PROGRAMMING

EPROM and Configuration Registers

PROM programming equipment makes the 87C75PF as easy to program as EPROM-version microcontrollers and standard EPROMs. Optimized programming equipment that utilizes the Quick-Pulse Programming™ algorithm can program the 87C75PF in less than four seconds.

Data I/O's model 29B (version V06), with Unipak-2B module (version 16, family/pin code = 112/107) and 87C75PF cartridge, supports the 87C75PF. It has a straightforward programming procedure. Assembled code is transferred to programmer RAM addresses 0000h 7FFFh. Configuration registers (CLR=7FFDh, ELR=7FFEh, and SFRLR=7FFFh) are loaded into programmer RAM addresses 8000H, 8001h, and 8002h. Configuration register contents can be entered manually using the programmers edit command.

With EPROM and configuration register contents loaded, the programmer automatically programs the EPROM array and non-volatile registers. The programmer can also read a programmed master device's EPROM array and configuration registers and program duplicates without further editing. Contact Data I/O or your programmer vendor for further details.

80C51 In-system Programming

Factory programmed and field updated applications use in-system and board programming techniques. Board programming equipment supplies voltages, addresses, data, and pertinent control signals to the board's edge-card connector.

In-system programming, on the other hand, allows a resident ROM- or EPROM-type microcontroller to program the system's off-chip non-volatile memory. A small amount of the microcontroller's ROM or EPROM contains code that controls its serial communications channel and knows how to program external EPROM.

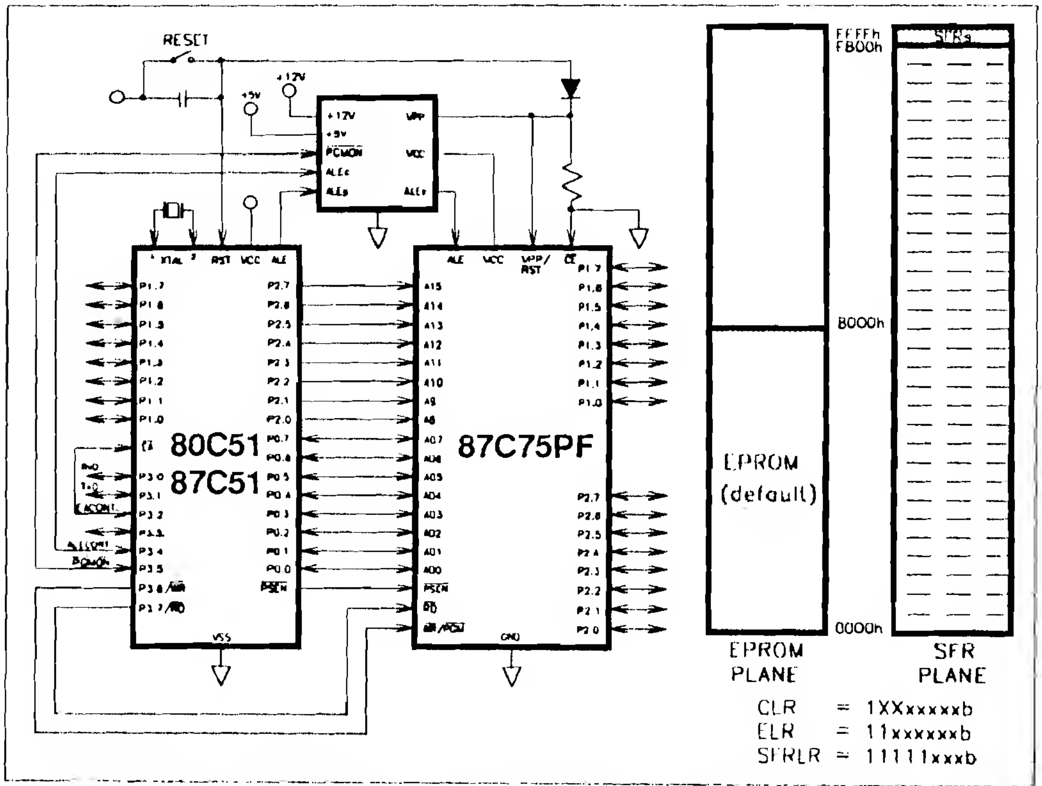

Figure 25. A simple circuit allows the microcontroller to program the 87C75PF In-system

Multiple-application modules can be customized using in system programming. For example, a generic control module can be built, installed in a variety of end products, and customized for different tasks at the end of the production sequence.

Figure 25 shows a simple 80C51-based in-system-programmable module. The microcontroller's on-chip ROM or EPROM contains the communication and programming algorithms. Port pins P3.0 and P3.1 provide the serial communication link. P3.2 (EACONT) controls the $\overline{EA}$ pin. When high (which occurs at reset or when "1" is written to it), internal program memory supplies code. When low, external EPROM supplies code. P3.4 (ALECONT) controls the ALE latching signal during programming. P3.5 ($\overline{PGMON}$) controls programming and operating mode V_{PP} and V_{CC} voltages. P3.6, which is the $\overline{WR}$ signal during normal operation, serves as the program pulse strobe, $\overline{PGM}$, during programming. $\overline{RD}$, P3.7, or $\overline{PSEN}$ can be used to verify programmed data whenever V_{PP} is at its programming voltage.

Figure 26 shows the program and latch control circuit. 5 volt and 12 volt supplies are connected to this circuit at all times. Inverter 74'06a allows 12 volts to pass into the DC/DC converter and the LM317 voltage regulators only when system power is on. $\overline{PGMON}$ is high after reset or when P3.5 contains a "1." $\overline{PGMON}$ controls inverter 74'06b which turns V_{PP} on or off. Inverter 74'06c keeps V_{CC} at 5 volts until programming commences. When $\overline{PGMON}$ goes low, these inverters turn off allowing V_{PP} and V_{CC} voltages to attain their programming levels. The variable resistors adjust V_{PP} and V_{CC} read- and program-voltages. V_{CC} read voltage is 5.0V and its program voltage is 6.25V. V_{PP} read voltage is 0V, so it doesn't interfere with the 87C75PF's reset, and its program voltage is 12.75V.

$\overline{PGMON}$ also controls the ALE circuit. When $\overline{PGMON}$ is high, the microcontroller's ALE value passes to the 87C75PF's ALE pin. When $\overline{PGMON}$ is low, the microcontroller's ALECONT controls ALE.

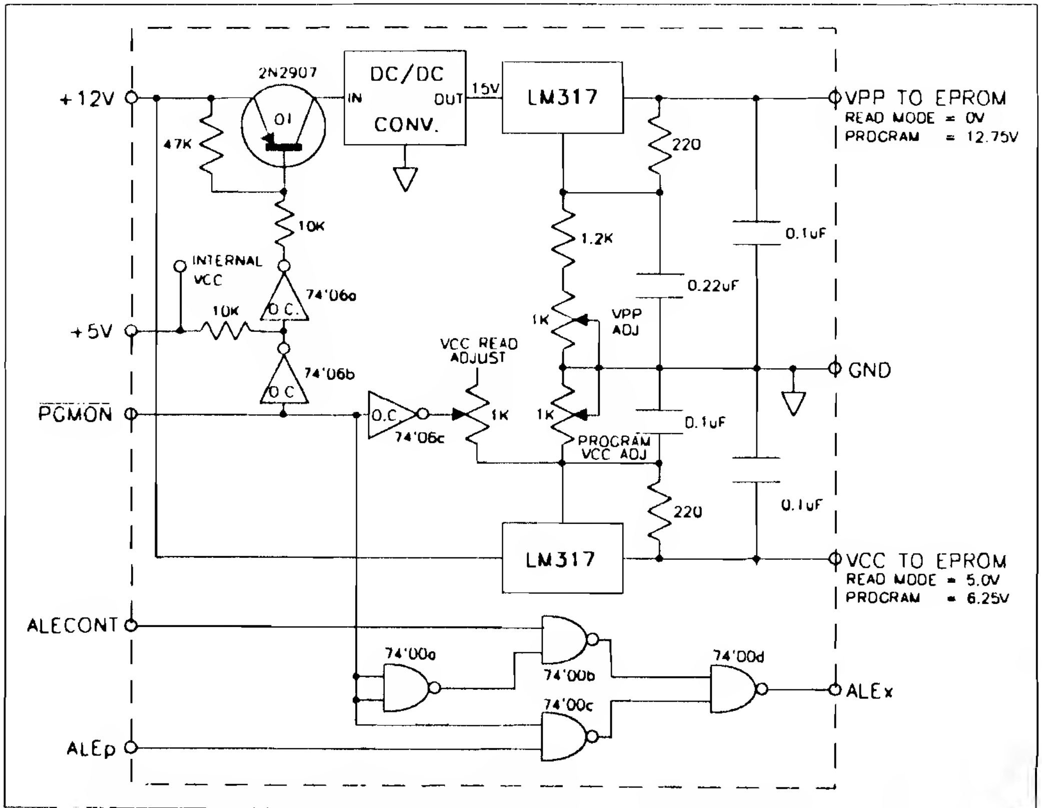

Figure 26. A microcontroller can use this circuit to control programming voltages and ALE.

The microcontroller's AD_{0-7} and A_{8-15} (ports 0 and 2) connect to the 87C75PF's AD_{0-7} and A_{8-15} pins. The controller's program-memory read signal, $\overline{PSEN}$, controls the 87C75PF's output-enable, $\overline{PSEN}$.

During programming, the controller brings EACONT high and $\overline{PGMON}$ low. This allows it to operate from internal code, enables programming voltages on the 87C75PF's V_{PP} and V_{CC} pins, and switches ALE control from the controller's ALE to its ALECONT. It then inputs data over its serial channel. With ALECONT high, an address is placed on ports 0 and 2. When ALECONT is brought low, the 87C75PF internally latches the address. Data read from the serial port is written to port 0. The Port Expander now has both address and data information. The controller needs only to bring its $\overline{WR}$ pin low to program data into the addressed location.

The in-system programming sequence is summarized below.

1) Set EACONT="1". Code is now supplied from the controller's internal program memory.

2) Assert $\overline{PGMON}$. This switches V_{PP} and V_{CC} to their program voltages and allows the controller to manually control ALE via ALECONT. ALECONT and $\overline{WR}$ are high.

3) Down-load address and data information via Port 3's serial channel. Ports 0 and 2 serve as I/O ports, so place the 16-bit address on them. Bring ALECONT low to latch the address into the 87C75PF.

4) Write data information to port 0.

5) Bring $\overline{WR}$ low to program data into the 87C75PF. See the 87C75PF data sheet for the programming algorithm and timing requirements.

6) Verify the programmed data. When the 87C75PF's V_{PP} is at 12.75V, its $\overline{PSEN}$ and $\overline{RD}$ pins are internally combined. The "MOVC A,@A+DPTR" instruction uses the $\overline{PSEN}$ pin to read EPROM data (or the MOVX A,@DPTR" instruction uses the $\overline{RD}$ pin).

7) Repeat this sequence until all EPROM data is programmed and verified.

8) When programming is complete, de-assert $\overline{PGMON}$ and ALECONT. When EACONT="0", code execution commences from the 87C75PF. Code duplication at identical internal and external memory locations allows uninterrupted paging between these two memory spaces.

When 6.25V is applied to the 87C75PF's V_{CC} during programming, its port outputs, when "1", will be close to 6.25V. Careful system design should ensure that microcontroller and other device inputs can handle this elevated voltage. Writing "0s" to all port pins before V_{CC} receives 6.25V will prevent damage to external devices.

SUMMARY

System demands push single-chip microcontroller designs to their limits. Complex applications are I/O intensive and use lots of EPROM. Traditional solutions use discrete chips — EPROM, address latches, address decoders, I/O port chips, and "glue" logic — to get more memory and expand, or recover, I/O.

Intel's 87C75PF Port Expander puts port functions, EPROM, and "glue" into a single package. Chip count and board size are dramatically reduced. System performance is optimized. Reliability is assured. Design time is shortened. Manufacturing is simplified. Device inventory is reduced.

Miniaturized system designs that weren't possible before, can now come to life, thanks to the 87C75PF.

intel

DOMESTIC SALES OFFICES

ALABAMA

†Intel Corp.
5015 Bradford Dr., #2
Huntsville 35805
Tel: (205) 830-4010

ARIZONA

†Intel Corp.
11225 N. 28th Dr.
Suite D-214
Phoenix 85029
Tel: (602) 869-4980

†Intel Corp.
1161 N. El Dorado Place
Suite 301
Tucson 85715
Tel: (602) 299-6815

CALIFORNIA

†Intel Corp.
21515 Vanowen Street
Suite 116
Suite 210
El Segundo 90245
Tel: (213) 640-6040

†Intel Corp.
1510 Arden Way, Suite 101
Sacramento 95815
Tel: (916) 920-8096

†Intel Corp.
4350 Executive Drive
Suite 105
San Diego 92121
Tel: (619) 452-5880

†Intel Corp.*
400 N. Tustin Avenue
Suite 450
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114

†Intel Corp.*
San Tomas 4
2700 San Tomas Expressway
2nd Floor
Santa Clara 95051
Tel: (408) 986-8086
TWX: 910-338-0255
FAX: 408-727-2620

COLORADO

†Intel Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (719) 594-6622

†Intel Corp.
650 S. Cherry St., Suite 915
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

CONNECTICUT

†Intel Corp.
26 Mill Plain Road
2nd Floor
Danbury 06811
Tel: (203) 748-3130
TWX: 710-456-1199

FLORIDA

†Intel Corp.
6363 N.W. 6th Way, Suite 100
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407
FAX: 305-772-8193

†Intel Corp.
5850 T.G. Lee Blvd.
Suite 340
Orlando 32822
Tel: (305) 240-8000
FAX: 305-240-8097

†Intel Corp.
11300 4th Street North
Suite 170
St. Petersburg 33716
Tel: (813) 577-2413
FAX: 813-578-1607

GEORGIA

†Intel Corp.
3280 Pointe Parkway
Suite 200
Norcross 30092
Tel: (404) 449-0541

ILLINOIS

†Intel Corp.*
300 N. Martingale Road, Suite 400
Schaumburg 60173
Tel: (312) 310-8031

INDIANA

†Intel Corp.
8777 Purdue Road
Suite 125
Indianapolis 46268
Tel: (317) 875-0623

IOWA

Intel Corp.

KANSAS

†Intel Corp.
10985 Cody St.
Suite 140, Bldg. D
Overland Park 66210
Tel: (913) 345-2727

MARYLAND

†Intel Corp.*
7321 Parkway Drive South
Suite C
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

†Intel Corp.
7833 Walker Drive
Suite 550
Greenbelt 20770
Tel: (301) 441-1020

MASSACHUSETTS

†Intel Corp.*
Westford Corp. Center
3 Carlisle Road
2nd Floor
Westford 01886
Tel: (617) 692-3222
TWX: 710-343-6333

MICHIGAN

†Intel Corp.
7071 Orchard Lake Road
Suite 100
West Bloomfield 48322
Tel: (313) 851-8096

MINNESOTA

†Intel Corp.
3500 W. 80th St., Suite 360
Bloomington 55431
Tel: (612) 835-6722
TWX: 910-576-2867

MISSOURI

†Intel Corp.
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990

NEW JERSEY

†Intel Corp.*
Parkway 109 Office Center
328 Newman Springs Road
Red Bank 07701
Tel: (201) 747-2233

†Intel Corp.
280 Corporate Center
75 Livingston Avenue
First Floor
Roseland 07068
Tel: (201) 740-0111
FAX: 201-740-0626

NEW MEXICO

†Intel Corp.
8500 Menaul Boulevard N.E.
Suite B 295
Albuquerque 87112
Tel: (505) 292-8086

NEW YORK

Intel Corp.
127 Main Street
Binghamton 13905
Tel: (607) 773-0337
FAX: 607-723-2677

†Intel Corp.*
850 Cross Keys Office Park
Fairport 14450
Tel: (716) 425-2750
TWX: 510-253-7391

†Intel Corp.*
2950 Expressway Dr., South
Suite 130
[illegible]
Suite 2B
Hollowbrook Park
Wappingers Falls 12590
Tel: (914) 297-6161
TWX: 510-248-0060

NORTH CAROLINA

†Intel Corp.
5700 Executive Drive
Suite 213
Charlotte 28212
Tel: (704) 568-8966

†Intel Corp.
2700 Wycliff Road
Suite 102
Raleigh 27607
Tel: (919) 781-8022

OHIO

†Intel Corp.*
3401 Park Center Drive
Suite 220
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528

†Intel Corp.*
25700 Science Park Dr., Suite 100
Beachwood 44122
Tel: (216) 464-2736
TWX: 810-427-9298

OKLAHOMA

†Intel Corp.
6801 N. Broadway
Suite 115
Oklahoma City 73162
Tel: (405) 848-8086

OREGON

†Intel Corp.
15254 N.W. Greenbrier Parkway
Building B
Beaverton 97006
Tel: (503) 645-8051
TWX: 910-467-8741

PENNSYLVANIA

†Intel Corp.*
455 Pennsylvania Avenue
Suite 230
Fort Washington 19034
Tel: (215) 641-1000
TWX: 510-661-2077

Intel Corp.*
400 Penn Center Blvd., Suite 610
Pittsburgh 15235
Tel: (412) 823-4970

PUERTO RICO

†Intel Microprocessor Corp.
South Industrial Park
P.O. Box 910
Las Piedras 00671
Tel: (809) 733-8616

TEXAS

†Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

†Intel Corp.*
12000 Ford Road
Suite 400
Dallas 75234
Tel: (214) 241-8087
FAX: 214-484-1180

†Intel Corp.*
7322 S.W. Freeway
Suite 1490
Houston 77074
Tel: (713) 988-8086
TWX: 910-881-2490

UTAH

†Intel Corp.

VIRGINIA

†Intel Corp.
1504 Santa Rosa Road
Suite 108
Richmond 23288
Tel: (804) 282-5668

WASHINGTON

†Intel Corp.
155 108th Avenue N.E.
Suite 386
Bellevue 98004
Tel: (206) 453-8086
TWX: 910-443-3002

†Intel Corp.
408 N. Mullan Road
Suite 102
Spokane 99206
Tel: (509) 928-8086

WISCONSIN

†Intel Corp.
330 S. Executive Dr.
Suite 102
Brookfield 53005
Tel: (414) 784-8087
FAX: (414) 796-2115

CANADA

BRITISH COLUMBIA

Intel Semiconductor of Canada, Ltd.
4585 Canada Way, Suite 202
Burnaby V5G 4L6
Tel: (604) 298-0387
FAX: (604) 298-8234

ONTARIO

†Intel Semiconductor of Canada, Ltd.
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel: (613) 829-9714
TLX: 053-4115

†Intel Semiconductor of Canada, Ltd.
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel: (416) 675-2105
TLX: 06983574
FAX: (416) 675-2438

QUEBEC

†Intel Semiconductor of Canada, Ltd.
620 St. John Boulevard
Pointe Claire H9R 3K2
Tel: (514) 694-9130
TWX: 514-694-9134

†Sales and Service Office
*Field Application Location

intel

UNITED STATES, Intel Corporation
3065 Bowers Ave., Santa Clara, CA 95051
Tel: (408)765-8080

JAPAN, Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi, Ibaraki, 300-26
Tel: 029747-8511

FRANCE, Intel Paris
1, Rue Edison, BP 303, 78054 Saint-Quentin-en-Yvelines Cedex
Tel: (33) 1-30 57 70 00

UNITED KINGDON, Intel Corporation (U.K.) Ltd.
Pipers Way, Swindon, Wiltshire, England SN3 1RJ
Tel: (0793) 696000

WEST GERMANY, Intel Semiconductor GmbH
Seidlstrasse 27, D-8000 Muenchen 2
Tel: (89) 53891

HONG KONG, Intel Semiconductor Ltd.
10/F East Tower, Bond Center, Queensway, Central
Tel: (5) 8444-555

CANADA, Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8
Tel: (416) 675-2105

Printed in USA/G152/0688/3K/BL/LK
Memory Components